

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**Viện Điện tử - Viễn thông**

**---db&ac---**



**Báo Cáo Bài Tập Lớn VLSI**

***Đề tài:*** [**Performance Evaluation of Complex Multiplier Using Advance Algorithm**](http://www.ijecse.org/wp-content/uploads/2013/07/Volume-2Number-3PP-1018-1025.pdf)

**Giáo viên hướng dẫn: TS.Phan Xuân Vũ**

**Nhóm sinh viên thực hiện:**

|  |  |  |
| --- | --- | --- |
| **Họ và Tên** | **MSSV** | **Lớp** |
| Trần Văn Bách | 20182375 | ĐTVT-08 |
| Trần Tuấn Anh | 20182364 | ĐTVT-08 |
|  |  |  |
|  |  |  |
|  |  |  |

**Hà Nội, 11/2021**

# MỤC LỤC

[MỤC LỤC 2](#_Toc92693798)

[PHÂN CÔNG CÔNG VIỆC 2](#_Toc92693799)

[PHẦN 1: TOÁN HỌC VEDIC 3](#_Toc92693800)

[***1.1.*** ***Giới thiệu về toán học VEDIC*** 3](#_Toc92693801)

[***1.2*** ***Tổng quan sơ đồ khối*** 4](#_Toc92693802)

[***1.2.1*** ***Bộ cộng Half adder*** 5](#_Toc92693803)

[***1.2.2*** ***Bộ cộng Full adder*** 5](#_Toc92693804)

[***1.2.3*** ***Bộ cộng N bit*** 7](#_Toc92693805)

[***1.2.4*** ***Bộ trừ N bit*** 7](#_Toc92693806)

[***1.2.5*** ***Vedic 2x2*** 8](#_Toc92693807)

[***1.2.6*** ***Vedic 4x4*** 10](#_Toc92693808)

[***1.2.7*** ***Vedic 8x8*** 11](#_Toc92693809)

[PHẦN 2: KẾT QUẢ 12](#_Toc92693810)

[KẾT LUẬN 14](#_Toc92693811)

[**TÀI LIỆU THAM KHẢO** 15](#_Toc92693812)

# PHÂN CÔNG CÔNG VIỆC

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên** | **Công việc** |
| 1 | Trần Văn Bách | * Tìm hiểu Vedic mathematics * Thiết kế bộ cộng - trừ * Thiết kế bộ nhân Vedic |
| 2 | Trần Tuấn Anh | * Tìm hiểu Vedic mathematics * Thiết kế bộ cộng - trừ * Thiết kế bộ nhân Vedic |

# PHẦN 1: TOÁN HỌC VEDIC

## ***Giới thiệu về toán học VEDIC***

Toán học Vedic là tên được đặt cho một hệ thống toán đã được đưa ra ánh sáng bởi Bharati Krisna Tirathji năm 1911. Ông là một người nghiên cứu Vedas và các thánh thư khác để đưa ra một hệ thống toán học khác với toán học truyền thống nhưng mạnh hơn, đơn giản và hiệu quả hơn. Ông đã chứng minh rằng tất cả các vấn đề trong toán học hiện đại có thể được giải quyết thông qua 16 bài kinh cơ bản hoặc công thức. Đây không phải là công thức toán học nhưng tinh thần hướng dẫn học sinh theo phương pháp thích hợp để giải quyết một khoản tiền.

Toán học Vedic khác với toán học truyền thống theo nghĩa là nó là mạch lạc và thay vì các kỹ thuật khác nhau được sử dụng cho các vấn đề khác nhau, nó sử dụng tính liên kết chặt chẽ để giải quyết vấn đề. Toán Vedic đơn giản và làm cho một sinh viên thực hiện các bước liên quan đến việc giải quyết vấn đề tinh thần. Nó linh hoạt hơn và có hệ thống hơn toán học truyền thống và thay vì buộc học sinh phải tuân theo một phương pháp quy định; Toán Vedic khuyến khích một sinh viên tìm ra phương pháp giải quyết sáng tạo và duy nhất của riêng mình.

Có rất nhiều điều về Vedic maths ngày nay và cảm thấy giá trị của nó trong việc làm cho trẻ học toán học một cách tốt hơn và dễ dàng hơn, các nhà giáo dục đang ủng hộ nhiều nghiên cứu về toán học Vedic để một chương trình học có thể được phát triển để làm cho ngay cả những người sợ hãi toán học để có toán học và phát triển trình độ trong đề tài này.

Bộ nhân là khối chính trong việc thiết kế các hệ thống kỹ thuật số đã có rất nhiều kỹ thuật nhân nhanh được trình bày trong cáo báo cáo ngày nay. Chúng em sẽ đề cập đến thuật toán nhân Vedic để giúp cho việc nhân nhanh va hiệu quả và thuật toán này làm mất hệ số nhân Vedic. Có 3 phương pháp nhân trong toán học Vedic. Trong số đó chỉ có 1 phương pháp là được sử dụng cho cả ba và 2 phương pháp kia là 2 trường hợp đặc biệt. Thuật toán chính em đề cập đến đó là “**Urdhva Triyakbhyam** **Sutra**” nó được hiểu theo nghĩa đen là nhân theo chiều dọc và theo chiều ngang.

Phép nhân giữa 2 số hệ 10 “46” và “33” khi sử dụng thuật toán nhân Vedic đạt được bằng cách nhân theo chiều dọc và theo chiều ngang và sau đó thêm tất cả các kết quả. Chúng ta có thể hiểu như sau 2 số “46” và “33” có thể được đại diện là 46=(4x40+6) và 33=(3x30+3). Phép nhân 46x33 được biểu diễn là (3x6 + 40x3 + 30x6 + 30x40) phép nhân nay được hiển thị trong hình [1]

A picture containing table

Description automatically generated

*Hình 1 Phép nhân Vedic 2 số “46” và “33”*

Tương tự như vậy thuật toán này cũng có thể được áp dụng cho số nhị phân

## ***Tổng quan sơ đồ khối***

Diagram

Description automatically generated

*Hình 2: Sơ đồ khối nhân 2 số phức 8bit*

Để xây dựng được sơ đồ khối như bên trên chúng em phải đi từ bộ nhân vedic 2x2 sau đó xây dựng đến bộ nhân vedic 4x4 rồi sau đó đến bộ nhân vedic 8x8 và còn dùng các bộ hỗ trợ kể đến như bộ: cộng, trừ, half-full adder…sau đây sẽ là chi tiết từng giai đoạn

### ***Bộ cộng Half adder***

Mạch cộng half-adder là mạch tổ hợp thực hiện chức năng cộng giá trị hai ngõ vào không tính đến cờ nhớ. Ngõ ra mạch cộng là giá trị tổng và cờ nhớ sinh ra từ kết quả cộng. Mạch cộng half-adder một bit có bảng sự thật như sau:

A picture containing text, light

Description automatically generated

*Hình 3 Bảng sự thật bộ cộng Half adder*

Diagram, schematic

Description automatically generated

*Hình 4 Sơ đồ bộ cộng Half adder 1-bit*

### ***Bộ cộng Full adder***

Mạch cộng Full adder là mạch tổ hợp thực hiện chức năng cộng giá trị hai ngõ vào có tính đến cờ nhớ. Mạch công Full adder một bit có bảng sự thật như sau

A picture containing text, light, white

Description automatically generated

*Hình 5 Bảng sự thật bộ cộng Full adder*

Diagram, schematic

Description automatically generated

*Hình 6 Sơ đồ bộ cộng Full adder 1-bit*

### ***Bộ cộng N bit***

Một bộ cộng n-bit có cả cờ nhớ ngõ vào cin và cờ nhớ ngõ ra cout thì có thể được ghép tầng để tạo ra bộ công cho chuỗi bit dài hơn. Có 2 phương pháp cộng cơ bản

* Bộ cộng Ripple-carry
* Bộ cộng Carry look - Ahead (CLA)

Ở trong báo cáo này chúng em sử dụng Ripple-carry còn Carry look – Ahead chúng em xin phép không đề cập.

Bộ cộng Ripple-carry sử dụng một bộ cộng half-adder và n-1 bộ cộng full-adder để thực hiện một bộ cộng n-bit.

Diagram

Description automatically generated

*Hình 7 Sơ đồ bộ cộng N-bit*

Chúng ta thấy là các bộ cộng này nối tiếp nhau và cộng lần lượt từ phải qua trái. Bộ ở bên phải cùng là bộ Half adder vì là cộng bit đầu tiên nên không có cờ nhớ còn lại các bộ cộng sau là bộ cộng Full adder sử dụng cờ nhớ từ bộ trước đó.

Trễ là vấn đề của Ripple-carry nên nó ít được sử dụng trong thực tế.

### ***Bộ trừ N bit***

Tương tự như bộ cộng, để xây dựng nên bộ trừ n-bit chúng em sẽ thêm các cổng **ex-or** vào mỗi bộ cộng full adder và hình dưới sẽ minh họa. Khi giá trị M = 1, bộ sẽ hoạt động như một bộ subtractor.

Diagram

Description automatically generated

*Hình 8 Sơ đồ bộ trừ N-bit*

Ở đây sử dụng toàn bộ các bộ cộng full adder và ở mỗi bộ cộng ta thêm cổng ex-or.

Khi M = 1, mạch là một bộ trừ và khi M = 0, đoạn mạch trở thành một bộ cộng. Cổng Ex-OR bao gồm hai đầu vào mà một đầu vào được kết nối với B và đầu vào khác với đầu vào M.

* Khi M = 0, B Ex-OR của 0 tạo ra B. Sau đó, các bộ cộng đầy đủ thêm B với A với đầu vào mang số không. và do đó một hoạt động bổ sung được thực hiện.
* Khi M = 1, B Ex-OR của 0 tạo ra phần bổ sung B và cũng mang đầu vào là 1. Do đó các đầu vào B bổ sung được thêm vào A và 1 được thêm vào thông qua phần mang đầu vào, không gì khác ngoài hoạt động bổ sung của 2. Do đó, phép tính trừ được thực hiện.

### ***Vedic 2x2***

Như em nói ở trên thì dưới đây sẽ là thuật toán “**Urdhva Triyakbhyam** **Sutra**” áp dụng cho số nhị phân 2 bit “10” và “10” phép nhân được biểu diễn ở hình số [3]

Diagram

Description automatically generated

*Hình 9 Các bước nhân 2 số nhị phân 2 bit*

Ở đây thì chúng ta thấy được rõ các bước để nhân 2 số nhị phân 2-bit và chúng ta có thể suy ra được sơ đồ phần cứng mô tả cho hình [3] như sau

Diagram, schematic

Description automatically generated

*Hình 10 Sơ đồ khối bộ nhân vedic 2x2*

Trong đó “ a ” và “ b ” là 2 số thực được nhân q là output. Với thiết kế này chúng em đã sẵn sàng bắt tay vào code Verilog và sử dụng các bộ cộng nửa tổng HA. Để làm cho chuyên nghiệp hơn chúng em sẽ cố gắng viết code bộ HA trước và sau đó sử dụng nó để có sản phẩm cuối cùng

### ***Vedic 4x4***

Ở đây bộ vedic 4x4 sẽ sử dụng lại bộ vedic 2x2 và 3 bộ cộng(adder) chúng ta có thể tạo bộ nhân vedic 4x4 như hình dưới

Diagram

Description automatically generated

*Hình 11 Thiết kế bộ vedic 4x4 sử dụng bộ vedic 2x2*

Với việc khởi tạo bộ nhân và bộ cộng 2x2 trước tiên ta phải viết code cho bộ 4bit và 6bit trước khi viết bộ vedic 4x4 để có thể thiết kế đơn giản hơn, chúng em sẽ sử dụng toán tử “+ ” được hỗ trợ bởi công cụ tổng hợp XST, theo mặc định sẽ chọn giá trị thấp bộ cộng phần cứng . Kiến trúc này tuân theo cây Wallace làm giảm mức độ bổ sung từ 3 xuống 2 giai đoạn như thể hiện trên hình 3. Quá trình nhân sẽ được diễn tả như sau:

Chart, scatter chart

Description automatically generated

*Hình 12 Quá trình thực hiện phép nhân 4x4*

### ***Vedic 8x8***

Tương tự như bộ vedic 4x4 ở đây để thiết kế bộ vedic 8x8 ta sử dụng 4 bộ vedic 4x4

Diagram

Description automatically generated

*Hình 13* *Thiết kế bộ vedic 8x8 sử dụng bộ vedic 4x4*

Quá trình thực hiện phép nhân được diễn tả ở dưới

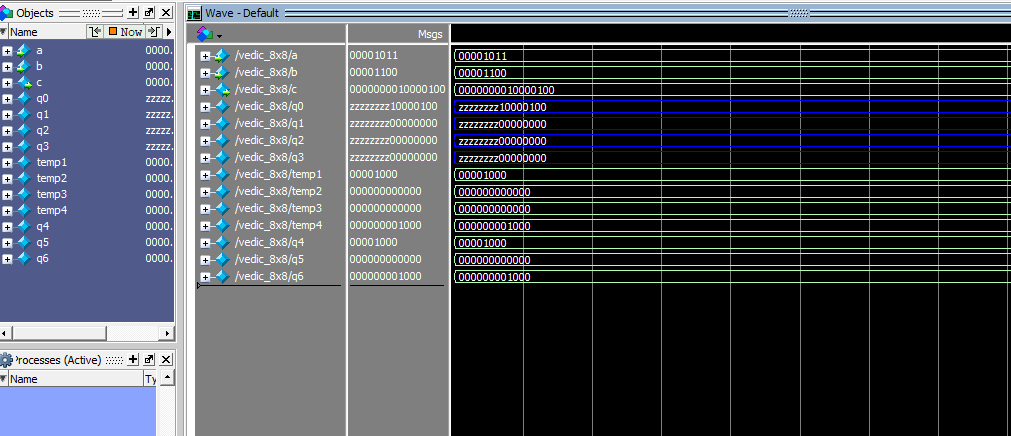
Diagram

Description automatically generated

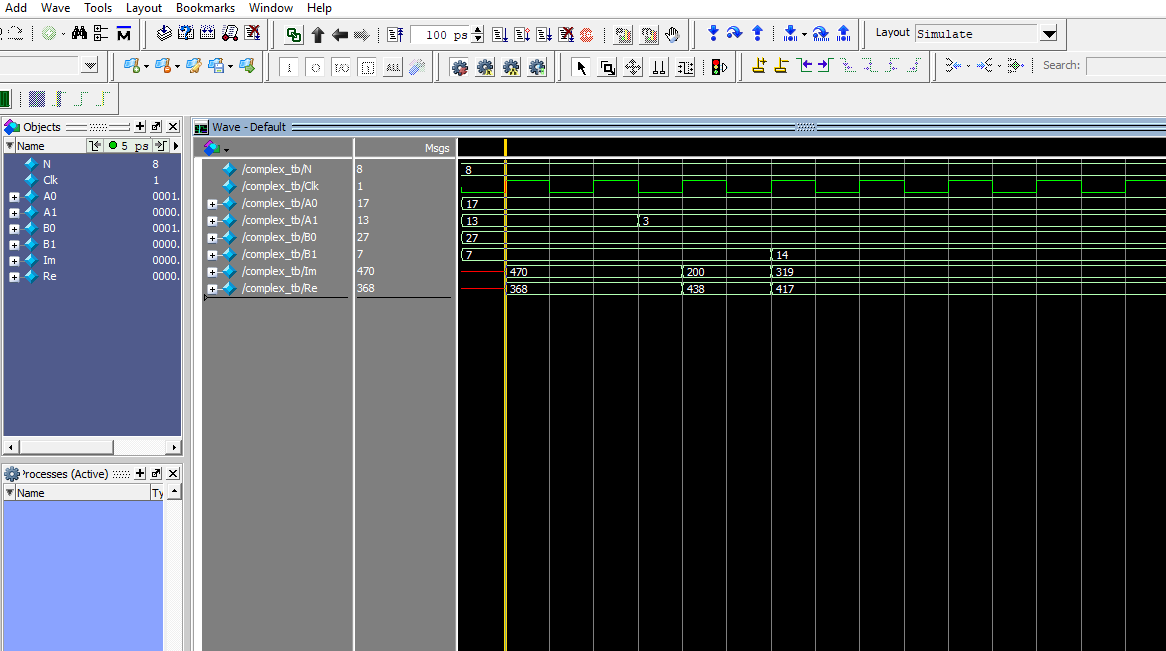
*Hình 14* *Quá trình thực hiện phép nhân 8x8*

# PHẦN 2: KẾT QUẢ

**1. Kết quả vedic 8x8:**



**2. Kết quả thực hiện bộ nhân phức:**



# KẾT LUẬN

Sau thời gian tìm hiểu, nghiên cứu và thực hiện đề tài dưới sự giúp đỡ, giảng dạy của thầy Phan Xuân Vũ, cùng với sự cố gắng, lỗ lực của hai thành viên trong nhóm đến nay đề tài đã hoàn thành và đã chạy mô phỏng chính xác kết quả so với tính toán lý thuyết.

Qua quá trình tìm hiểu và thực hiện đề tài, nhóm đã thu được kết quả chính như sau:

* Nghiên cứu, tìm hiểu bộ nhân số phức với thuật toán nhân vedic 8 bit.
* Thiết kế, mô phỏng bộ nhân vedic và bộ nhân phức.
* Biết sử dụng phần mềm mô phỏng Modelsim và biết sử dụng ngôn ngữ mô tả phần cứng Verilog.

Tuy kết quả mô phỏng chưa được tối ưu hóa hoàn toàn, do vậy chúng em rất mong nhận được sự nhận xét, đánh giá từ thầy để đề tài hoàn thiện hơn.

Một lần nữa chúng em xin chân thành cảm ơn sự giảng dạy và giúp đỡ của thầy Phan Xuân Vũ để chúng em hoàn thành đề tài.

**TÀI LIỆU THAM KHẢO**

[1] : [VEDIC Multiplier - Digital System Design](https://digitalsystemdesign.in/vedic-multiplier/#:~:text=Main%20algorithm%20of%20Vedic%20multiplication%20is%20Urdhva%20Triyakbhyam.,and%20Crosswise%20and%20then%20adding%20all%20the%20results.)

[2]: https://citeseerx.ist.psu.edu/viewdoc/downloaddoi=10.1.1.686.9296&rep=rep1&type=pdf

[3]: https://www.electronicshub.org/binary-adder-and-subtractor/?fbclid=IwAR1J7BuyekInn4D3hU8zYdKsZZ3DuY8NSFytg9qwnYfugz95LXMmu9NhmrA

[4] : http://verilog-code.blogspot.com/2014/01/design-and-implementation-of-16-bit.html

[5]: https://www.chipverify.com/verilog/verilog-tutorial